A Design Methodology for Modeling Trustworthy Value Webs

Jaap Gordijn and Yao-Hua Tan

ABSTRACT: In this paper we introduce a design methodology for business models from two perspectives: the value web perspective and the trust perspective. The value web perspective models the creation, distribution, and consumption of goods or services of economic value in a network of multiple enterprises and end-consumers. The goal of the methodology is to create a shared understanding of a business model for all actors involved, and to assess the potential profitability. The trust perspective describes how value webs can be expanded with trustworthy control procedures thus enhancing confidence of actors in each other to enable trading. We present a first outline of a formal theory to design trustworthy control procedures in the setting of the e^3 value methodology.

KEY WORDS AND PHRASES: network organisations, business models, design methodology, trust, control.

Introduction

e-Business development becomes more and more a *trans-disciplinary* design problem. One of the very first design issues concerns the business model. Various definitions of the notion of 'business model' exist (see [22] for a survey). Most definitions agree that a business model presents a way of doing business and does so from different perspectives. Typically, various stakeholders are involved in developing a business model; e.g. business analysts, marketing and sales experts, IT developers, management and even the board members (CxOs) of various enterprises. These different stakeholders face the same design problem, how to develop a specific business model, from different viewpoints. It is widely acknowledged that these different viewpoints result, due to misunderstandings amongst stakeholders, in a lack of shared understanding of the e-business model to be developed. Moreover, in the case of value webs consisting of various enterprises, stakeholders often represent different interests. Since enterprises often do not share a common terminology, e.g. because they operate in different markets and have other corporate cultures, shared understanding about an e-business case is even harder to obtain.

One of the contributions of the computer and management sciences is the *conceptual modeling* approach for business and information technology development. Conceptual modeling refers to defining aspects of the physical and social world with the purpose of improving the understanding of and communication about these aspects [21]. Modeling an e-business case may contribute to a better understanding of this and therefore increases the stakeholder's confidence in such a case.

Since there are multiple perspectives on an e-business model, it is important to use multiple-perspective modeling techniques to represent the different stakeholder viewpoints. A typical problem in conceptual modeling is to express all perspectives by one single description technique, which leads to unclear and ambiguous descriptions of a particular e-business case, which does not contribute to a shared understanding. In this paper we introduce modeling techniques for two of such perspectives: the *value web* perspective and the *trust* perspective.

The value web perspective models the creation, distribution, and consumption of economic value in a network of multiple enterprises and end-consumers. Our goal is two-fold: (1) to create a shared understanding of a business model for all actors involved, and (2) to assess the potential profitability of a business model. In this paper, we focus on shared understanding. We use the *e³value* methodology, which has been successfully used to model value webs in various industry sectors, such as the telecom, banking, energy and entertainment industry.

It is widely acknowledged that trust between trade partners in a value web is a key to the success of a business relation (see e.g. [1, 18, 30, 23 and 4]). In particular, in e-business relations, where parties often start online business with each other without having any previous experience with each other, or lack detailed information about one another, trust building is a complicated aspect of the relation (see e.g. [20]). For trust modeling in relation to value web modeling the following observations are important:

(1) *Trust services can be considered as viable commercial services themselves* that involve many enterprises (e.g. banks). In other words: trust services themselves are parts of value webs, which we can consider as *secondary* webs, in contrast to the *primary* webs (e.g. trading goods for money) that they facilitate. So, part of trust modeling could actually be done by using a technique such as *e³value* to specify trust facilitating value webs.
(2) *Trust services presuppose specific knowledge, beliefs and acceptance of obligations by the actors involved*. It is important to understand these notion, to be able to create confidence in the use of trust controls.
(3) *Trust services require trust controls*, which are often implemented as inter-organizational business processes and/or codified in information systems. These controls e.g. monitor the receipt of invoices, and check their correctness against other documents (e.g. a contract) provided by other controls. In case of exceptions, appropriate actions can be taken.

This paper focuses on the first two observations: how to model a value web offering trust services, and how to model assumptions regarding knowledge, beliefs and obligations related to such trust services. The third observation, modeling of controls, is a topic of further research. For example, in [27] a first outline how to model these controls can be found, which is based on applying techniques from colored Petri nets to *e³value*.

The remainder of this paper is structured as follows. Section 2 focuses on modeling value webs while section 3 proposes a way to model trust. In section 4, we elaborate on relating value web modeling to trust modeling. Finally, section 5 presents future research to be done to connect value web and trust modeling.

Modeling Value Webs in *e³value*

The *e³value* methodology [8] has been developed to model a value web consisting of actors who create, exchange, and consume things of economic value. It has been used to model value webs in various industries, e.g. the music, finance, internet service provisioning, news and energy industry [9]. Moreover, elementary tool-support is available (see http://www.cs.vu.nl/~gordijn/research.htm), and advanced tool support is currently developed in the EC-IST funded project Obelix (see http://obelix.e3value.com). In this section we give an overview of the main *e³value* methodology concepts, and subsequently we present a value web for a specific trust service, namely the Letter of Credit Procedure.

The e³value Methodology: Modeling Constructs

To represent an e-business value model an ontology was developed for *e³value* that consists of interrelated core concepts and the Use Case Maps scenario technique [2]. An advantage of the *e³value* ontology is that it contains a small set of concepts and relations. This makes it relatively easy for business consultants, business analysts, and CxO's to apply *e³value*. Moreover, the agility of e-business projects (the need to rapidly define, explore, and execute a business idea [10]) requires such an approach. For the same reason, *e³value* is based on an executable scenario technique. A number of other business oriented ontologies have been developed [29, 6, 7], but they all contain too many constructs to be useful for conceptual modeling in practice. For example, Malone's ontology [17] provides ontology constructs for "inventing organizations". This ontology consists of 3400 different activities with 20 levels of specializations and 10 decomposition levels. Such a complex ontology is hardly suitable for exploring new networked propositions in practical business situations. One reason for this complexity is that Malone's ontology (and many others) focuses on *business processes* ('how') rather than on *value propositions* ('what'). In contrast, by focusing on the value proposition, the *e³-value* methodology requires only a small number of ontological constructs and allows users to abstract away from operational details of the actual business processes that are used to execute the value proposition.

The *e³value* modeling constructs differ from techniques provided by the Unified Modeling Language (UML) [24] in various respects. The UML has activity diagrams but these diagrams represent the sequence of an object flow. The *e³value* methodology does not model *sequences* but represents *dependencies* between objects of economic value. Dependencies do not assume any time ordering. Additionally, the UML does not contain the notion of objects of economic value and consequently does not provide the appropriate terminology for discussing value webs. Finally, the UML does not have the notion of value interface to model economic reciprocity. It is not directly possible to model that to obtain something, one should offer something else in return.

To support understanding of *e³value* models, they are represented graphically (see Figs. 1, 3 and 4). Experiences with business users in various research projects have shown that this is a particular useful feature of *e³value*. Here we briefly explain the ontological concepts and the UCM scenario concepts as well as the profitability assessment. More detailed information about *e³value* can be found in [8, 9].

Figure 1 gives an example of a simple value model, representing that a shopper receives a good, and pays money in return. The shop obtains goods from a store, which purchases them from a wholesaler. The wholesaler obtains goods from a manufacturer. Note that we only model actors exchanging objects of *economic value*, and no other interactions between actors such as exchange of control information about the business processes.

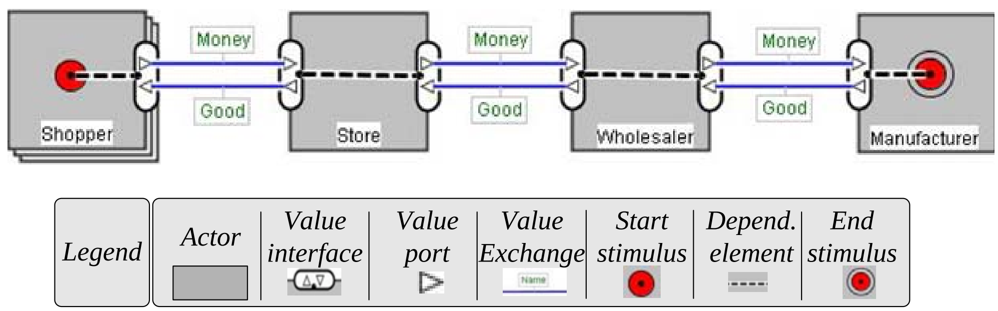

Figure 1: An example *e³value* model

(Note: The Legend is only for explanatory purposes and are not part of the *e³value* modeling technique itself)

Fig. 1 consists of the following *e³value* base constructs:

- **Actor.** An actor is perceived by its environment as an independent economic (and often also legal) entity. By performing *value activities* (see below) an actor makes profit or increases its utility. In a sound and viable business value model *every* actor should be able to make a profit. Actors are represented as rectangles. Store, Wholesaler, and Manufacturer are examples of actors. Shopper is visualized as stacked actors, denoting a *market segment*. In our interpretation of market segment, actors in a segment attribute equal economic value to objects. The decision to model an entity as an actor or as a segment is determined by the modeling and analysis purpose: e.g. in Fig. 1 the motivation can be that we are interested in analyzing potential profit for a chain of companies in relation to an end-customer market segment.
- **Value Object.** Actors exchange value objects. A value object can be a service, right, good or even a consumer experience. The important point is that a value object represents a *value* for one or more actors. Value objects are shown as text next to arrows. Value objects in Fig. 1 are Good and Money.
- **Value Port.** An actor uses a value port to show to its environment that it wants to provide or request value objects. The concept of a port is important, because it enables to abstract away from the internal business processes, and to focus on how external actors and other components of the e-business value model can be 'plugged in'. Ports are shown as small black circles.
- **Value Interface.** Actors have one or more value interfaces. A value interface consists of individual value ports offering or requesting value objects. It shows the value object(s) an actor is willing to exchange *in return for* other value object(s). Such willingness is expressed by a decision function on the value interfaces, which shows under what conditions an actor wants to exchange a

value object for another value object. The exchange of value objects is *atomic* at the level of the value interface; i.e. either *all* exchanges occur as specified by the value interface or *none* at all. Note that a value interface does not indicate the temporal ordering of objects to be exchanged on its ports. It only indicates which value object is available, in return for some another value object. A value interface is shown by a rounded box, connected to an actor. In Fig. 1, value interfaces denote that actors offer/request a good and request/offer money in return.

- **Value Exchange.** A value exchange is used to connect two value ports with each other. A value exchange represents one or more *potential* trades of value objects between value ports. As such, it is a prototype for actual trades between actors. According to the Enterprise Ontology [29] a value exchange would be called a potential sale. It shows which actors are willing to exchange value objects with each other. A value exchange is shown by an arrow.

Paths are used to relate an actor's value interfaces. Whereas value exchanges show *inter*-actor dependencies, paths represent *intra*-actor dependencies. They show via which value interface(s) an actor must exchange value objects, given the exchange of objects via another interface of that same actor. The main purpose of paths is to facilitate the *counting* of value exchanges in an entire value web, as a result of a consumer need. This facilitates a profitability analysis on a per actor basis (see for more details [9]). Note that our paths do not represent *time-ordering*. Paths are only used to present *dependencies* between value exchanges of objects via value interfaces. For representation of paths, a simple form of Buhr's Use Case Maps [2] is used. The two constructs used to build a path are *dependency elements* and *connection elements*. Connection elements interconnect dependency elements like value interfaces, resulting in paths.

- **Dependency element.** A path is expressed by dependency elements, interconnected by connection elements. Essentially, a path gives dependencies between value interfaces (a kind of connection element) so that we can reason for an entire value model what happens with other value interfaces if we exchange values via one particular value interface. Dependency elements are denoted by dotted lines.
- **Connection element.** A connection element connects various dependency elements. Connection elements can be start or stop stimuli, AND/OR forks or joins and value interfaces. Different types of connection elements are represented differently.
- **Stimulus element.** Paths start with one or more *start stimuli*. A start stimulus represents an event, possibly caused by an actor. In most cases, such a stimulus represents a consumer need. A start stimulus has a *counter* representing the number of events per time frame. If a start stimulus is directly connected via a dependency element to a value interface, the counter denotes how many times value objects are exchanged via that interface during a specific time frame. Dependency elements as well as connection elements all have a counter, but the counter for a start stimulus should be given by the modeler. Counts for other elements can be derived by a traversing the path and propagating the count. If an actor causes an event, the start stimulus is represented in the actor box. A path also has one or more *end stimuli*. They have no successors. A start stimulus is

represented by a bulls-eye, an end-stimulus is represented by a bulls-eye with a surrounding circle.

- **AND** and **OR connection elements.** An *AND* element connects a dependency element d_1 to one or more other dependency elements $d_2 \ldots d_n$. The purpose is to propagate the counter of a dependency element d_1 over dependency elements $d_2 \ldots d_n$. The counters for dependency elements $d_1 \ldots . d_n$ are all the same. An *OR* element models a continuation of the path into one direction, to be chosen from a number of alternatives. It connects a dependency element d_1 to one or more other dependency elements $d_2 \ldots d_n$, but now the counter of dependency element d_1 is distributed over dependency elements $d_2 \ldots d_n$ using a given distribution ratio. An *AND* element is shown as a line, perpendicular to the lines visualizing dependency elements. An *OR* element is represented by a triangle.
- **Value interface revisited.** Another way to connect dependency elements is to use a value interface. We use value interfaces (connected by dependency elements) and its associated counter to create profitability sheets on a per actor basis to assess profitability. Such a sheet shows the objects of value that are exiting or entering an actor as a result of path execution, caused by a stimulus. This is shown in Fig.1. If a store sells a good, it also has to buy the same good from the wholesaler. This dependency between the two value interfaces of the store is represented by the dependency element inside the Store actor box.

With these constructs a conceptual model of value web can be constructed. Value webs typically consist of multiple enterprises, represented by multiple stakeholders. The *e*[3]*value* modeling constructs help to create a shared understanding of the value web. In addition to this ontology-based graphical design tool *e*[3]*value* also supports the calculation of the economic benefit of a value model via the so-called profitability assessment (see for full details [9]). It consists of two main steps: (1) profitability sheet generation, and (2) evolutionary scenario-based assessment.

A profitability sheet shows for an actor an economic value-based quantification of the in-going and out-going value objects, based on the number start-stimuli (modeling consumer needs) per time period, say per month. To generate such sheets, so-called *valuation* formulas have to be specified for objects representing money, and preferably also for all objects (at least end-consumers have to evaluate the non-monetary objects they obtain). Additionally, the number of start-stimuli has to be estimated. Given these numbers, profitability sheets can be automatically generated in *e*[3]*value*. Each sheet shows for an actor the net cash flow (in case of end-customers this includes also values for non-monetary objects). Fig. 2 is an example of a profitability sheet for the model in Fig. 1.

	A	B	C	D	E	F
1	**Value Interface**	**Value Port**	**Value Exchange**	**Occurrences**	**Valuation**	**Economic Value**
2	Buy store	total for Buy store		10000		-900000
3		Good	(all connected)	10000	0	0
4		Payment	Money	10000	90	-900000
5	Sell store	total for Sell store		10000		1000000
6		Payment	Money	10000	100	1000000
7		Good	(all connected)	10000	0	0
8						
9	total for actor			20000	0	100000
10						

value_interface / value_exchange / **Store** / Wholesaler

Figure 2: An example profitability sheet

Profitability sheets are particularly useful to do a sensitivity analysis of the expected net cash flow. Not only estimates about the number of start-stimuli and the valuation formulas themselves are subject to uncertainty, but also the value web itself, including the actors and the value exchanges, may evolve over time. Consequently, evolutionary scenario techniques known from strategic decision making [11] are used to investigate the sensitivity of the expected cash flow with respect to future events. This sensitivity analysis enhances the understanding of the value web, and many stakeholders indicated that this was even more insightful than the profitability numbers themselves.

Case Study: Letter of Credit

As case study we use the Letter of Credit procedure. Banks introduced this procedure in order to solve the following problem in international trade. Suppose we have a seller in Hong Kong and a buyer in the Netherlands. The agents are geographically far apart, and the goods have to be transported by a carrier from the seller to the buyer (we assume over sea). On the one hand the seller does not want to ship the goods onto the carrier's vessel (and thereby lose control over them) without first receiving payment from the buyer. On the other hand the buyer does not want to pay the seller (and thereby lose control over the money) before the goods have been shipped. In other words, the agents prefer a simultaneous exchange of the shipment of the goods in return for the money. To solve this deadlock situation banks introduced the letter of credit, which is an agreement that the bank of the buyer, the so-called issuing bank, will arrange the payment for the seller as soon as the seller can prove to the bank that he shipped the goods. The seller proves this shipment by presenting the Bill of Lading to the bank, the so-called corresponding bank. The seller receives the bill of lading from the carrier, when the seller shipped the goods. The seller's bank transfers the Bill of Lading to the customer's bank and the customer's bank gives the Bill of Lading to the customer as soon as the customer pays. Finally, the customer receives the shipped goods from the carrier in return for the bill of lading. The bill of lading is an example of a multimodal transport document that has an evidentiary effect. This evidentiary effect is even stipulated in a special convention of the United Nations. The *United Nations Convention on International Multimodal Transport of Goods* (CIMTG) describes this function as follows [28]:

> **Article 10 -** *Evidentiary effect of the multimodal transport document*
>
> *Except for particulars in respect of which and to the extent to which a reservation permitted under article 9 has been entered:*
>
> *The multimodal transport document shall be prima facie evidence of the taking in charge by the multimodal transport operator of the goods as described therein; and*
> *Proof to the contrary by the multimodal transport operator shall not be admissible if the multimodal transport document is issued in negotiable form and has been transferred to a third party, including a*

consignee, who has acted in good faith in reliance on the description of the goods therein.

This is a case study for which trust controls themselves are well known (see observation (3) in Section 1). Our value and trust modeling tools should help to understand the assumptions regarding knowledge, belief and obligations and stakeholders, and the trust service value web. This can be seen as a case of reverse engineering, whereas in most cases we start with the design of a trust value web, then express knowledge, beliefs and obligations assumptions and finally design trust controls themselves.

Primary Value Web: Exchanging Goods for a Fee

The Letter of Credit procedure can be considered from multiple perspectives. Seen from a trust perspective, the Letter of Credit procedure contributes to increasing confidence in reliable and fair exchanges of goods between actors, who do not know each other in advance. From a value model perspective, the Letter of Credit procedure can be seen as a commercial service *itself* facilitating the sale and delivery of a good or service. Actually, the letter of credit is a commercial service, because the buyer has to pay a fee to the bank that issues the letter of credit. If a value web is considered as a set of actors performing economic exchanges with each other, we can view the Letter of Credit procedure as an economically valuable service in a *secondary* value web, facilitating a *primary* value web consisting of actors exchanging goods or services. In Fig. 3 a primary value web is represented, modeling that a supplier offers some object of value to a customer and obtains a fee in return.

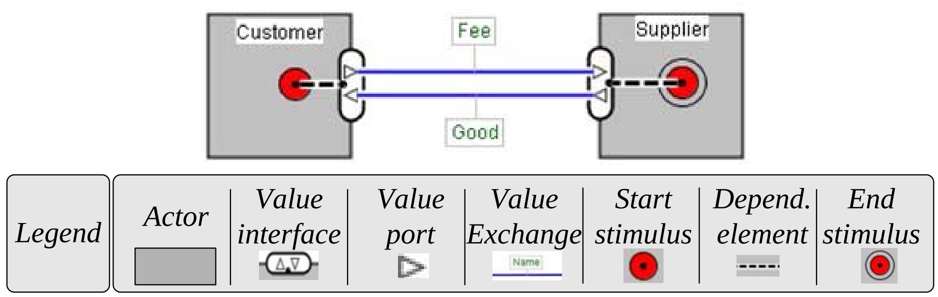

Figure 3: A supplier and a customer exchanging objects of value.

It is important to understand that, given the semantics of the *e^3value* concepts, this value model states that a supplier is only willing to provide a good *if and only if* it obtains a fee (of course, the reverse requirement holds for the customer). In other words, the supplier is only willing to exchange objects via *all* ports of its value interface, or *none* at all. Hence, in this value model economic reciprocity is assumed to hold. *How* this is ensured is not an issue when designing a value web in the first place; then we focus only on the value proposition itself.

Secondary Value Web: Letter of Credit

The value web presented in Fig. 3 has an important normative statement. We state that if the supplier delivers a good to a customer, he always gets paid. The same holds for the consumer: if he pays, he obtains the good. More generally, value interfaces model economic reciprocity; if you deliver something of value to someone, you get something else of value in return for that, and vice versa. To understand and analyze value webs, this is precisely what we want to model.

However, in practice the norm of economic reciprocity does not always hold. For example, a customer might order and receive a good, but then refuse to pay the supplier. This means that the principle of economic reciprocity is violated. Hence, control mechanisms are required, to ensure that both value exchanges in Fig. 3 occur (or none at all). The Letter of Credit is such a control mechanism, which can also be regarded as a commercial service itself. Hence, in Fig. 4 the *primary* service of selling a good is expanded with a kind of *secondary* control service, the Letter of Credit procedure, which is specifically tailored to secure the interests of the seller.

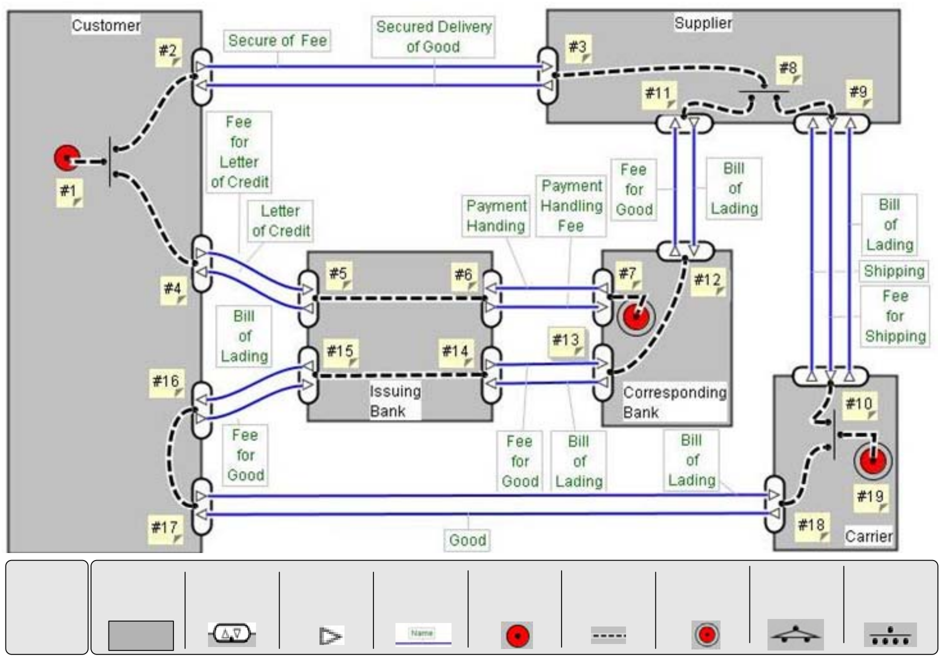

Figure 4: Secondary value web for Letter of Credit.

The Letter of Credit procedure considered from a commercial service perspective is represented in Fig. 4 and represents that the *customer* must guarantee that the *supplier* gets paid for the good. This is depicted by the AND-fork (a kind of connection element, see #1), indicating that if the consumer wants a good, he must exchange values via interfaces #2 and #3 (a good for a fee) *and* via interfaces #4 and #5. The latter is the obtainment of a Letter of Credit, a service which ensures that if the supplier ships a good, then he gets paid. The customer obtains a Letter of Credit from an *issuing bank* and the customer pays a fee for this to the issuing bank. Typically, the issuing bank of the Letter of Credit is in the same country as the customer, but often has no branch in the supplier's country. In that case the issuing bank needs to collaborate with a *corresponding bank*, which is physically close to the supplier. This corresponding bank pays the supplier, when the supplier presents the bill of lading to the bank as evidence that he has shipped the good. In return for this service, which is an intrinsic part of the Letter of Credit procedure, the corresponding bank charges the issuing bank a fee (see interfaces #6 and #7). Note that the Letter of Credit, of which the supplier is notified by the issuing bank, is a guarantee for the supplier that he will be paid. This is reflected in the value web by the exchange *secure of fee*, rather than just *fee*. A secured fee means that the issuing bank *guarantees* that the seller will be paid as soon as he can prove shipment of the good. Also note that there is no direct value exchange between the issuing bank and the supplier; the aforementioned notification is seen as part of business process that we do not represent in value models. We only represent its effect, which is a secured fee.

As a result of securing the fee, the supplier exchanges objects of value via two of its interfaces, represented by the AND-fork #8. Via interfaces #9 and #10, the supplier ships the ordered good via a *carrier*. The carrier charges the supplier a shipping fee, and the supplier obtains a Bill of Lading from the carrier. It is important to understand that the Bill of Lading has an economic value. It is a so-called negotiable document, which can itself be traded and is similar in this respect to paper money. Additionally, from a trust perspective, it is important to understand that the carrier is seen as a trusted third party. All actors involved in the letter of credit assume that the carrier only gives a Bill of Lading if he obtains the good to be shipped from the supplier. As soon as this Bill of Lading is presented by the supplier to the corresponding bank, then the bank pays the fee for the good to the supplier. Hence, this is a kind of secured pre-payment arrangement for the seller. So, via interfaces #11 and #12, the supplier offers the Bill of Lading, obtained from the carrier to the corresponding bank, and in return obtains a fee for the good. Note that Fig. 4 only shows dependencies between value exchanges, not the temporal ordering of actual events.

Subsequently, the Bill of Lading is transferred by the corresponding bank to the issuing bank (interfaces #13 and #14). As a consequence, the issuing bank exchanges with the customer the Bill of Lading for the fee for the good (interfaces (#15 and #16). The carrier transports the good to the customer, and releases the good to the customer, in return for the Bill of Lading (interfaces #17 and #18). The AND join, annotated with #19, models that the Bill of Lading, as issued by the carrier, should also be obtained by the carrier once the good is delivered. After that, the Bill of Lading has served its purposes.

Analyzing Trust in Control Procedures

An e^3*value* model represents the principle of economic reciprocity. It takes the mechanism of 'one good turn deserver another' as normative behavior. However, in real-life, actors will commit frauds sometimes and violate the principle of economic reciprocity. Consequently, it is important to enhance a business model with trust mechanisms and control procedures.

An important way to create trust in a control mechanism is to understand how it works, and how the controls protect you against opportunistic behavior of a trading partner in a commercial transaction (see e.g. [3]). In this section we analyze how trust is created by control procedures by formalizing what it means to understand a control procedure for trust. This is based on earlier work that was introduced in [25, 26].

According to Article 10 of the CIMTG the Bill of Lading as shipment document *reliably indicates* that the goods have been shipped in international trade procedures. Note that this article has a normative element. Whether the Bill of Lading is evidence does not depend so much on whether a person is psychologically convinced by it, but the law simply stipulates that everybody involved in a letter of credit procedure *should* consider this document as sufficient evidence. We use the conditional operator $\Rightarrow_P$, which denotes 'reliably indicates', to formalize the following so-called evidence rule:

$$\text{BoL} \Rightarrow_P \text{Shipped} \quad \text{(Evidence Rule)} \qquad (1)$$

This is read as 'in the context of procedure P, the Bill of Lading reliably indicates that the goods were shipped'. (The axiomatization of this and other formal notions below can be found in the Formal Appendix.)

We described the evidence rule and the procedure in an objective manner, i.e. in terms of objective facts such as 'BoL' and 'Shipped'. For the actual execution of the procedure, however, the mental states of the agents involved are equally important. If one of the agents does not believe the facts, or something went wrong, e.g. an agent did not receive the Bill of Lading, then the procedure does not work. Hence, we cannot simply use objective facts like 'BoL' and 'Shipped' for modeling the mental states of the agents, but we have to use subjective beliefs about such facts to model the mental state of the agents. To model these belief states of agents, we use epistemic operators such as $B_i(\varphi)$, which denotes that agent i believes φ, and $K_i(\varphi)$, which denotes that agent i knows φ. If agent b believes the Bill of Lading, then we represent this by the formula $B_i(\text{BoL})$. This belief depends on the agent's belief that the document is not forged, e.g. because the document comes from a trustworthy source. Similarly, $B_i(\text{Shipped})$ means that the agent i believes that shipment of the goods took place. We model the fact that an agent understands the evidence rule of the procedure with the following formula.

$$K_i(B_i(\text{BoL}) \Rightarrow_P B_i(\text{Shipped})) \quad \text{(Epistemic Evidence Rule)} \qquad (2)$$

This formula says that agent i knows that, if he believes the Bill of Lading, then according to the procedure P he has a good reason to believe that the goods are shipped. Note the importance of the procedural setting here. Agent i knows that by law he should consider the bill of lading as sufficient evidence for shipment. We use here the knowledge operator, because procedures or legal texts are non-empirical information (like the rules of a game, or mathematics). In other words, these are not empirical data about which you can make incorrect observations. You either know this or not, but there is nothing in between, while you can get misleading information about empirical facts such as a bill of lading or shipment. Hence, we use the B_i operator to represent the belief in these facts.

It is a general principle in most legal systems that its norm subjects are supposed to know the laws. This can be represented by an obligation for the norm subjects to 'ought to know'. The norm in Article 10 expresses that everybody who uses the letter of credit procedure *ought to know* that the Bill of Lading reliably indicates that the goods were shipped, which can be formalized as follows.

$$O_i(K_i(B_i(BoL) \Rightarrow_P B_i(Shipped))), \text{ for all agents } i \quad (3)$$

(Obligatory Knowledge of the Evidence Rule)

The idea of rule (3) is that an agent cannot use the argument that he did not know rule (2) as justification for violating an obligation. In other words, if agent i has received the Bill of Lading but did not pay for the goods on time, then agent i cannot justify his violation of the obligation to pay for the goods by saying that he did not know that the goods have been already shipped. Agent i ought to have known that receiving the Bill of Lading reliably indicates that the goods were shipped.

The last issue that we have to address is the *shared knowledge* aspect of trust. The letter of credit procedure is only trustworthy for the seller if he knows about all the other parties involved, e.g. the buyer, the buyer's bank etc, that they will accept the bill of lading as proof of shipment. In other words, the seller has to know that the evidence rule is obligatory knowledge for all the other parties. We model this by the following formula:

$$K_i(O_j(K_j(B_j(BoL) \Rightarrow_P B_j(Shipped)))) \text{ for all agents } j \neq i \quad (4)$$

The understanding of the procedure is modeled by the fact that the rules 2, 3 and 4 are part of the agent's mental state. In other words, the formula $B_i(BoL)$ by itself only models that the agent believes the written content of the Bill of Lading. However, understanding the complete control functionality of the Bill of Lading requires a lot more. The agent has to know (1) all the legal consequences and functional roles of the Bill of Lading, and (2) the agent has to know what he and other agents are supposed to know about these consequences. This second part is also called common knowledge (see [5]); i.e. knowing what all the other agents know, and knowing that the others also have this knowledge about yourself. In particular, this common knowledge is essential for trust creation between the agents of a trading community.

Value Modeling and Trust Analysis

In *e*3*value* modeling atomicity of value interfaces is assumed. For example, in the purchase scenario it is assumed that the buyer always pays for the goods that are delivered. However, in real life the seller has often a major concern whether the buyer will indeed pay after the goods have been delivered. In Section 2 we used the Letter of Credit example to illustrate *how* a particular control procedure can have the effect of reducing these concerns for the seller. In Section 3 we showed how our formal analysis of trust could explain *why* the Letter of Credit is effective in trust creation. To apply *e*3*value* modeling not just for business model development of value webs, but also for the actual implementation of the organizational and governance aspects of value webs, it is useful to extend *e*3*value* with a set of secondary control value webs. Secondary control value webs are commercial trust services that can be added to the initial business model design in order to obtain sufficient trust levels for all actors to actually do transactions. The ultimate research objective is to create a library of secondary control *e3value* webs templates, or short control templates that can be applied to various business models. However, designing these control templates is a complicated task, because business models can vary in many different aspects. For example, the Letter of Credit is an example of a control template that is useful for most high-value infrequent (international) purchasing situations. However, in the case of very frequent low-value transactions, a Letter of Credit is typically too expensive. An example of frequent low-value economic exchanges is the economic exchanges in the telecom industry between telecom operator and Internet service provider, where for each Internet activity of the customer of an ISP a small sum is paid by the ISP to the telecom operator. In this case other control templates than Letters of Credit are more suitable. In this case a so-called 'equity hostage exchange' or joint venture could be more appropriate (see [31, 4]). Another reason why the design of control templates is complicated is that in the case of electronic commerce many traditional control mechanisms could become obsolete. For example, the Bill of Lading could become obsolete if an information system would be introduced that could track and trace all container movements (e.g. by having a GSM/GPS or Radio Frequency Identity (RFID) chip attached to every container). In that case the buyer would know the exact moment when the goods are shipped by the seller to the carrier without needing the Bill of Lading, and similarly for the corresponding bank that does the actual payment. Hence, the Letter of Credit control template could be redesigned for this situation. In particular, the Bill of Lading becomes obsolete. Note, however, that this IT solution only works if the buyer is convinced that the system is run by an independent third party that is not dominated by the seller. For example, a telecom company could provide this independent tracking and tracing information system as a commercial service.

Our analysis, how trust is created by control procedures by formalizing what it means to understand a control procedure, can be used to support this design and redesign of control templates. By analyzing the belief states and lack of belief or knowledge of all parties involved in a business model, one gets a better picture of the concerns and uncertainties of each of the parties. Based on this precise information about the concerns one can select a control template, or perhaps even design a new one. For example, select a Letter of Credit procedure if there is no advanced and independent tracking and tracing system available, or switch to a more efficient electronic procedure without Bill of Lading. In the latter case the crucial observation is that there are more efficient ways to create the belief for the buyer that the goods are shipped, i.e. the formula $B_b(\text{Shipped})$. This trust formalization can be viewed as a kind of formal requirements analysis for the design of control templates. Every control template for international transactions should satisfy the requirement that $B_b(\text{Shipped})$ is fulfilled before the Buyer or any other party acting on his behalf (as the issuing bank in the LC procedure) will make a payment. By taking this abstract perspective it is easier to redesign or even obliterate operational details of existing control templates. However, this formal trust analysis does not automatically generate ways to design control procedures. It only specifies the abstract requirements for such a control procedure, and it can be used to explain how a given control procedures creates trust. Trust analysis as requirements analysis is a useful prerequisite for the actual design of control procedures, just as in software development (formal) requirements analysis is an important preparation for the actual functional design of software.

Conclusions

Value models and trust models are two perspectives on an e-business case. A value model shows which actors are involved and which objects of economic value are exchanged between these actors. The *e^3value* methodology presupposes that these exchanges always occur according to the principle of economic reciprocity. In other words, there are no actors who are committing a fraud, or other mishaps which may result in failed exchanges of value.

Consequently, to realize e-business model into practice, mechanisms need to be added, which ensure that committed exchanges of value actually occur. In general, such mechanisms will be based on control procedures. Two interesting observations were made about these control procedures. First, trust-increasing control procedures themselves can be seen as viable commercial value-added services with a corresponding value model. We called such value models secondary, because they facilitate the exchange of values in another, primary, value model. Secondly, a theory is needed about control procedures and how to design them for specific value models. Just as the design methodology requires principles for the design of the primary value models, it also requires control specific principles for the design of the secondary trust services value models. Here we made a first attempt to develop such a theory for the design of the secondary trust services value models, based on the modeling of knowledge, beliefs and obligations of the actors. In future research we will further develop this theory, and based on this theory we will further develop the *e^3value* methodology with a library of heuristic guidelines for selecting the most appropriate control procedures for a given value web.

Formal Appendix

The knowledge operator K_i and belief operator B_i have the usual axioms and inference rules (see [5]).

K_i Axioms:
a) $K_i(\varphi \rightarrow \psi) \rightarrow (K_i\varphi \rightarrow K_i\psi)$
b) $K_i\varphi \rightarrow \varphi$
c) $K_i\varphi \rightarrow K_iK_i\varphi$
d) $\neg K_i\varphi \rightarrow K_i\neg K_i\varphi$

B_i Axioms:
a) $B_i(\varphi \rightarrow \psi) \rightarrow (B_i\varphi \rightarrow B_i\psi)$
b) $B_i\varphi \rightarrow \neg B_i\neg\varphi$
c) $B_i\varphi \rightarrow B_iB_i\varphi$
d) $\neg B_i\varphi \rightarrow B_i\neg B_i\varphi$

and the following inference rules:

a) if φ and $\varphi \rightarrow \psi$, then ψ
b) if $|- \varphi$, then $|- K_i\varphi$ and $|- B_i\varphi$

The conditional $A \Rightarrow_P B$, which denotes ‘A reliably indicates B’, has the following axioms as defined in [14]

$$(A \Rightarrow_P B) \wedge (A \Rightarrow_P C)) \rightarrow (A \Rightarrow_P (B \wedge C))$$

$$((A \Rightarrow_P B) \wedge (C \Rightarrow_P B)) \rightarrow ((A \vee C) \Rightarrow_P B)$$

and the following inference rules

If $|- A \leftrightarrow B$, then $|- (C \Rightarrow_P A) \leftrightarrow (C \Rightarrow_P B)$

If $|- A \leftrightarrow B$, then $|- (A \Rightarrow_P C) \leftrightarrow (B \Rightarrow_P C)$

Note that the modus ponens inference rule does not hold for $\Rightarrow_P$, hence it is much weaker than material implication. The evidence relation is not a causal relation. The rule, $BoL \Rightarrow_P Shipped$, does not mean that the bill of lading causes shipment, just as smoke does not cause fire.

The deontic operator O_i is the standard deontic logic (SDL) operator. We have the usual SDL axioms and inference rules (see e.g. [19]).

O_i Axioms:
a) $O_i(\varphi \rightarrow \psi) \rightarrow (O_i\varphi \rightarrow O_i\psi)$
b) $O_i\varphi \rightarrow \neg O_i\neg\varphi$

and the following inference rules:

a) if φ and $\varphi \rightarrow \psi$, then ψ
b) if $|- \varphi$, then $|- O_i\varphi$

Acknowledgements

This work has been partly sponsored by the EU-IST project IST-2001-33144 Obelix and the EU-EESD NNE5/2001/256 project BusMod.

References

1. Bons, R.W.H.; Lee, R.M.; and Wagenaar, R.W. Designing trustworthy inter-organizational trade procedures for open electronic commerce. *International Journal of Electronic Commerce*, 2, 3 (1998), pp.61-83.
2. Buhr, R. J. A. Use case maps as architectural entities for complex systems. *IEEE Transactions on Software Engineering*, 24, 12 (1998), pp. 1131–1155.
3. Das, T.K. and Teng, B.S. Between trust and control: developing confidence in partner cooperation in alliances. *Academy of Management Review*, 23, 3 (1998), pp. 491-512.

4. Dyer, J.H. and Singh H. The relational view: cooperative strategy and sources of interorganizational competitive advantage. *Academy of Management Review*, 23, 4 (1998), pp. 660-679.
5. Fagin, R.; Halpern, J.; Moses, Y.; and Vardi, M. *Reasoning About Knowledge*. Cambridge: MIT Press, 1995.
6. Fox, M.S. and Gruninger, M. Enterprise modelling. *AI Magazine*. 19, 3 (Fall 1998), pp. 109-121.
7. Geerts, G. and McCarthy, W.E. An accounting object infrastructure for knowledge-based enterprise models. *IEEE Intelligent Systems and Their Applications*. (July-August 1999), pp. 89—94.
8 . Gordijn, J. and Akkermans, J.M. Designing and evaluating e-business models. *IEEE Intelligent Systems - Intelligent e-Business*, 16, 4 (2001), pp. 11-17.
9. Gordijn, J., and Akkermans, J.M. Value based requirements engineering: exploring innovative e-commerce idea. *Requirements Engineering Journal*. Springer Verlag, 8, 2 (2003), pp. 114-134.
10. Hartman, A.; Sifonis, J.; and Kador, J. *Net Ready - Strategies for Success in the E-conomy*. New York: McGraw-Hill, 2000.
11. Heijden, K. van der *Scenarios: The Arts of Strategic Conversation*. New York: John Wiley & Sons Inc., 1996.
12. Holland, C.P. and Lockett, A.G., Business trust and the formation of virtual organizations. In Sprague, R.H. (ed.), *Proceedings of the 31st Annual Hawaii International Conference on System Sciences.* Los Alamitos: IEEE Computer Society, 1998.
13. Jasper, R. and Uschold, M. A framework for understanding and classifying ontology applications. In Gaines, B.; Cremer, R.; and Musen, M. (eds.), *Proceedings 12th Int. Workshop on Knowledge Acquisition, Modelling, and Management*, volume I, Banff: SRDG Publications, 1999, pp. 4–9–1 — 4–9–20.
14. Jones, A.J.I. and Sergot, M.J. A formal characterisation of institutionalised power. *Journal of the Interest Group in Pure and Applied Logics*, 4, 3 (1996), pp. 429-445.
15. Kotler, P. *Marketing Management: Analysis, Planning, Implementation and Control*. Englewood Cliffs: Prentice Hall, 1988.
16. Lee, R.M. and Bons, R.W.H. Soft-coded trade procedures for open-edi. *International Journal of Electronic Commerce*. 1, 1 (1996), pp. 27-49.
17. Malone, T. W.; Crowston, K.; Pentland, B.; Dellarocas, C.; Wyner, G.; Quimby, J.; Osborn, C.S.; Bernstein, A.; Herman, G.; Klein, M.; O'Donnel, E. Tools for Inventing Organizations: Towards a Handbook of Organizational Processes. *Management Science*, 45, 3 (1999) pp. 425-433.
18. Mayer, R.C.; Davis, J.H.; and Schoorman, F.D. An integrative model of organizational trust. *Academy of Management Review*. 20, 3 (1995), pp. 709-734.
19. Meyer, J.-J.Ch. and Wieringa, R.J. *Deontic Logic in Computer Science*. Chichester: John Wiley, 1993.
20. McKnight, D.H. and Chervany, N.L. What trust means in e-commerce customer relationships. *International Journal of Electronic Commerce*. 6, 2 (2002), pp. 35-53.
21. Mylopoulos, J. Conceptual modeling and Telos. In Loucopoulos, P. (ed.), *Conceptual Modelling, Databases and CASE: An Integrated View of Information Systems Development*. New York: Wiley, 1992, pp. 49—68.
22. Pateli, A.G. and Giaglis, G.M. A framework for understanding and analysing e-business models. In Wigand, R.T.; Tan, Y.H.; Gricar, J.; Pucihar, T. and Lunar,

T. (eds.), *Proceedings of 16th Bled Electronic Commerce Conference: eTransformation*, Kranj: Moderna Organizacija, 2003, pp. 329-348.
23. Rousseau, D. M.; Sitkin, S. B.; Burt, R. S.; and Camerer, C.F., (eds.). Special Issue on Trust in Organizations. *Academy of Management Review*, 23, 3 (1998).
24. Rumbaugh, J.; Jacobson, I.; and Booch, G. *The Unified Modelling Language Reference Manual*. Reading: Addison Wesley Longmann, Inc., 1999.
25. Tan, Y.H. and Thoen, W. Toward a Generic Model of Trust for Electronic Commerce. *International Journal of Electronic Commerce*. 5, 2 (2000), pp. 61-74.
26. Tan, Y.H. and Thoen, W. Formal aspects of a generic model of trust for electronic commerce. *Journal of Decision Support Systems*. 33, (2002), pp. 233-246.
27. Tan, Y.H.; Thoen, W.; and Gordijn, J. Modeling Controls for Value Exchanges in Virtual Organizations. *Proceedings of the 2nd Conference on Trust Management*, Berlin: Springer Verlag, 2003, pp. 236-250.
28. United Nations Convention on International Multimodal Transport of Goods. http://www.jus.uio.no/lm/un.multimodal.transport.1980/index.html, Geneva: United Nations, accessed on February 26, 2004.
29. Uschold, M.; King, M.; Moralee, S.; and Zorgios, Y. The Enterprise Ontology. *The Knowledge Engineering Review*. 13, 1, (1998), pp. 31–89.
30. Uzzi, B. Social structure and competition in interfirm networks: the paradox of embeddedness. *Administrative Science Quarterly*. 42, (1997), pp. 35-67.
31. Williamson, O.E. Credible commitments: using hostages to support exchange. *American Economic Review*. 73, (1983), pp. 519-535.

JAAP GORDIJN (gordijn@cs.vu.nl) is an assistant professor in e-Business at the Faculty of Exact Sciences of the Free University Amsterdam. His research interest is innovative e-business applications. He is a key developer of and has internationally published on the e^3-*value* e-business modelling methodology, addressing the integration of strategic e-business decision making with ICT requirements and systems engineering. Before joining the Free University he was a member of Cisco´s Internet Business Solution Group and senior manager of the e-Business group at Deloitte & Touche. As such, he was involved in rolling out e-business applications in the banking, insurance and digital content industry.

YAO-HUA TAN (ytan@feweb.vu.nl) is professor of electronic business at the Department of Economics and Business Administration of the Free University Amsterdam. His research interests are the role of trust and control in inter-organisational systems for dynamic network organisations and electronic commerce, ICT-enabled electronic contracting, and artificial intelligence techniques to enable automation of business procedures in international trade.